УДК 519.71

О ВЕРИФИКАЦИИ КОНЕЧНЫХ ПАРАМЕТРИЗОВАННЫХ МОДЕЛЕЙ РАСПРЕДЕЛЕННЫХ ПРОГРАММ

П.Е. БУЛЫЧЕВ
В.А. ЗАХАРОВ

Московский государственный университет им. М.В. Ломоносова

e-mail: zakh@cs.msu.su

В настоящей работе введен и исследован новый класс параметризованных моделей распределенных программ. Модели представляют собой конечные размеченные системы переходов, в которых в качестве параметров используются булевы переменные. Благодаря этому открывается возможность компактного описания конечных семейств взаимодействующих процессов, в которых оценки событий в различных состояниях и осуществимость переходов между состояниями определенным образом согласованы между собой. Предложен символьный алгоритм верификации конечных параметризованных моделей программ для логики деревьев вычислений CTL, а также установлены оценки сложности рассматриваемой задачи.

Ключевые слова: программа, система переходов, спецификация, темпоральная логика, верификация, булева функция, OBDD, NP-полная задача.

Одной из актуальных задач системного программирования является проверка правильности сложных программ. Каждое средство верификации программ создается на основе специальной математической теории, в рамках которой определяются модели программ, язык описания требований правильного поведения программ (спецификации), а также алгоритмы проверки выполнимости этих требований для моделей программ. Наиболее часто в качестве языка спецификаций используется язык формальной логики, а в роли модели проверяемой программы выступает одна из логических интерпретаций (моделей) этого языка, в которой представлены всевозможные вычисления этой программы. Тогда задача проверки правильности поведения программы сводится к хорошо известной задаче математической логики – проверке выполнимости заданной логической формулы в заданной модели (model checking, верификация моделей программ). Для многих классов программ, вычисления которых проявляются во взаимодействии их компонентов между собой и с окружающей средой, верификацию моделей программ наиболее удобно проводить на основе темпоральных логик. Моделью программы в этом случае является размеченная система переходов (модель Крипке), в состояниях которой дается оценка осуществимости событий, и задача верификации программы сводится к задаче проверки выполнимости заданной темпоральной формулы на заданной системе переходов. Хорошо известны эффективные алгоритмы проверки выполнимости темпоральных формул в конечных моделях (см. [1]), сложность которых пропорциональна размеру модели. На основе этих алгоритмов были созданы системы верификации программ nuSMV [2], SPIN [3] и др., применяемые при разработке программного обеспечения и микроэлектронных схем.

Но существуют и такие распределенные программные системы, для описания которых приходится использовать бесконечные семейства моделей со сколь угодно большим числом состояний. Семейства моделей программ такого рода называются параметризованными моделями программ. Параметром модели обычно служит какая-либо характеристика программы, например, количество процессов, емкость каналов связи, максимальный размер данных и т. п. Как было показано в статье [4] , в общем случае задача верификации параметризованных моделей программ алгоритмически неразрешима. Тем не менее, для отдельных классов распределенных систем, применяемых на практике, удалось разработать специальные методы, позволяющие проводить верификацию соответствующих параметризованных моделей программ. К числу основных методов верификации параметризованных моделей распределенных программ относятся метод выявления симметрии [5-9], метод абстракции [10-12], метод инвариантов [13-15]. При помощи этих методов удалось успешно проверить правильность некоторых распределенных алгоритмов и сетевых протоколов.

В настоящей работе введен и исследован новый класс параметризованных моделей программ, представляющих собой конечные размеченные системы переходов, в которых в качестве параметров используются булевы переменные. Все переходы системы, а также события в каждом состоянии системы помечены булевыми формулами, зависящими от переменных-параметров. Для каждого набора значений параметров булевы формулы, помечающие события, определяют оценку событий во всех состояниях системы. При этом доступными считаются лишь те переходы, которые помечены булевыми формулами, выполнимыми на заданном наборе значений переменных-параметров. Поскольку множество наборов значений булевых параметров конечно, каждая такая параметризованная модель определяет конечное множество примеров обыкновенных (непараметризованных) моделей программ. Благодаря этому открывается возможность компактного описания конечных семейств моделей программ, в которых оценки событий в различных состояниях и осуществимость переходов между состояниями определенным образом согласованы между собой.

При помощи переменных-параметров можно описывать разнообразные требования, налагаемые на топологию коммуникационной подсистемы, а также вводить зависимость между осуществимостью событий в разных состояниях системы. Рассмотрим, например, следующую задачу маршрутизации. Предположим, что распределенная вычислительная система состоит из n процессов, соединенных ненадежными каналами связи. Необходимо убедиться, что адаптивный алгоритм маршрутизации обеспечивает доставку сообщений из одного узла этой системы в другой узел всякий раз, когда исправные каналы связи образуют связную сеть. Чтобы проверить это условие корректности, можно проанализировать по отдельности все возможные варианты неисправности каналов связи, при которых сохраняется связность коммуникационной сети. Применяя методы верификации моделей программ, для каждого варианта неисправности сети можно проверить выполнимость требования неизбежной доставки сообщения. Однако если количество вариантов неисправности сети очень велико (возрастает экспоненциально с увеличением числа узлов сети), то такой способ проверки правильности распределенного алгоритм практически неприемлем. Альтернативный способ решения этой задачи верификации программ позволяет ограничиться изучением одной-единственной конечной параметризованной модели программ. Для этого достаточно ввести вспомогательные булевы параметры и пометить каналы связи булевыми функциями, зависящими от этих параметров. Система булевых функций, помечающих каналы связи, должна быть выбрана таким образом, чтобы

1) для любой связной подсети существовал набор значений параметров, на котором значение true принимают только те булевы функции, которые приписаны каналам связи этой подсети;

2) для любого набора значений параметров подсеть, образованная всеми теми каналами связи, у которых соответствующие булевы функции принимают значение true на этом наборе, являлась связной.

Для выбора подходящей разметки можно применить алгоритмы теории тестирования управляющих систем [16]. Как только указанная разметка каналов связи сформирована, мы получаем конечную параметризованную модель распределенной системы, для которой можем проверять выполнимость требования неизбежной доставки сообщений.

Конечные модели подобного рода исследовались в работе [17]. Авторы этой работы использовали трехзначную логику для описания моделей с частично определенной структурой. В предложенной нами разновидности конечных параметризованных моделей допускаются гораздо более широкие возможности использования параметров, нежели в работе [17]. Нами также предложен символьный алгоритм верификации конечных параметризованных моделей программ для логики деревьев вычислений CTL, а также установлены оценки сложности рассматриваемой задачи.

Вначале мы коротко напомним основные понятия темпоральной логики деревьев вычислений CTL, затем введем параметризованные модели программ (модели Крипке), и в завершении опишем символьный алгоритм верификации конечных параметризованных моделей Крипке и оценим его сложность.

1. **Темпоральная логика CTL.** Темпоральная логика деревьев вычислений (Computational Tree Logic, CTL) была предложена для формального описания требований корректного поведения, которые предъявляются к реагирующим вычислительным системам. К числу систем такого рода относятся интерактивные, многонитевые, распределенные программы, встроенные информационные системы. Реагирующим системам присущи две характерные особенности – недетерминизм и незавершаемость вычислений. Поэтому поведение реагирующей системы определяется множеством ее бесконечных вычислений, организованном в виде дерева. В каждом состоянии вычисления могут быть зарегистрированы те или иные события, например, отправление запроса, открытие доступа к ресурсу, изменение значения переменной и др. Каждому из таких событий сопоставляется атомарная формула, принимающая логическое значение *true* только в том случае, когда по ходу вычисления реагирующей системы осуществляется соответствующее событие. Формулы CTL описывают причинно-следственные зависимости между событиями, происходящими в различные моменты времени в процессе функционирования реагирующей системы. При помощи формул CTL можно задавать требования корректности, определяющие желаемое поведение системы, а затем для заданного формального описания (программы, схемы, спецификации и др.) реагирующей системы проверять, используя методы, алгоритмы и инструментальные средства верификации моделей программ, выполнимость этих требований. Строгое определение синтаксиса и семантики CTL таково.

Пусть задано множество $A = \{a_1, a_2, ..., a_n, ...\}$ атомарных формул (событий). Формулой CTL называется всякое выражение φ, которое либо является атомарной формулой, либо является составной формулой одного из перечисленных ниже видов:
$\neg\psi$, $\psi_1 \wedge \psi_2$, $\psi_1 \vee \psi_2$, $\forall X\psi$, $\exists X\psi$, $\forall F\psi$, $\exists F\psi$, $\forall G\psi$, $\exists G\psi$, $\forall(\psi_1 U \psi_2)$, $\exists(\psi_1 U \psi_2)$,

где ψ, ψ_1, ψ_2 – формулы CTL. Логические связки $\neg$ (отрицание), $\wedge$ (конъюнкция) и $\vee$ (дизъюнкция) имеют тот же смысл, что и в классической логике. Темпоральные операторы X («в следующий момент времени», neXt_time), F («когда-нибудь», some time in Future), G («всегда», Globally) и U («до тех пор пока», Until) служат для описания осуществимости событий на протяжении бесконечного вычисления реагирующей системы. Кванторы $\forall$ и $\exists$ указывают, относится ли утверждение об осуществимости событий во времени ко всем вычислениям системы или только к одному из вычислений. Например, формула $\forall(\neg(a_1 \wedge a_2)U(\exists F a_0))$ утверждает, что в каждом вычислении системы события a_1 и a_2 не могут осуществляться совместно, до тех пор пока не будет достигнуто некоторое состояние, начиная из которого вычисление может быть продолжено таким образом, что рано или поздно осуществится событие a_0.

Семантика (интерпретация) формул CTL определяется на основе размеченных систем переходов (моделей Крипке). Моделью Крипке называется четверка $M = (S, S_0, R, L)$, где S – непустое конечное множество состояний, S_0 – подмножество начальных состояний, $R \subseteq S \times S$ – множество переходов, $L : S \times A \to \{true, false\}$ – оценка событий. При этом предполагается, что отношение переходов является тотальным: для любого состояния s существует состояние s', в которое ведет переход из s, т.е. $(s, s') \in R$. Вычислением в модели Крипке M называется бесконечная последовательность состояний $\pi = s_1, s_2, \ldots, s_i, s_{i+1}, \ldots$, в которой для каждого номера $i, i \geq 1$, выполняется отношение перехода $(s_i, s_{i+1}) \in R$. Условимся использовать запись $\pi[i]$ для обозначения i-го состояния в вычислении π.

Для заданной модели Крипке M, ее состояния $s, s \in S$, и формулы CTL φ бинарное отношение выполнимости $M, s \models \varphi$ определяется следующим образом:

- Если $\varphi = a \in A$, то $M, s \models \varphi \Leftrightarrow L(s, a) = true$;
- Если $\varphi = \neg\psi$, то $M, s \models \varphi \Leftrightarrow M, s \not\models \psi$;

- Если $\varphi = \psi_1 \wedge \psi_2$, то $M,s \models \varphi \Leftrightarrow M,s \models \psi_1$ и $M,s \models \psi_2$;
- Если $\varphi = \psi_1 \vee \psi_2$, то $M,s \models \varphi \Leftrightarrow M,s \models \psi_1$ или $M,s \models \psi_2$;
- Если $\varphi = \forall X\psi$, то $M,s \models \varphi \Leftrightarrow$ для любого состояния s', в которое ведет переход из состояния s, верно $M,s' \models \psi$;
- Если $\varphi = \forall F\psi$, то $M,s \models \varphi \Leftrightarrow$ для любого вычисления π, исходящего из состояния s, существует такое $i, i \geq 1,$ для которого верно $M,\pi[i] \models \psi$;
- Если $\varphi = \forall G\psi$, то $M,s \models \varphi \Leftrightarrow$ для любого вычисления π, исходящего из состояния s, что для всякого $i, i \geq 1,$ верно $M,\pi[i] \models \psi$;
- Если $\varphi = \forall(\psi_1 U \psi_2)$, то $M,s \models \varphi \Leftrightarrow$ для любого вычисления π, исходящего из состояния s, существует такое $i, i \geq 1,$ что $M,\pi[i] \models \psi$ и для всякого $j, 1 \leq j < i,$ верно $M,\pi[j] \models \psi$.

Отношение выполнимости для формул $\exists X\psi$, $\exists F\psi$, $\exists G\psi$, $\exists(\psi_1 U \psi_2)$ определяется аналогичным образом с той лишь разницей, что вместо *всех* переходов и вычислений, исходящих из состояния s, достаточно рассмотреть лишь *один* переход или вычисление.

2. **Параметризованные модели Крипке.** В параметризованных моделях Крипке состояния и переходы систем помечены булевыми функциями, зависящими от параметров. Для каждого набора значений этих параметров значения этих функций определяют оценку атомарных формул и активность переходов. Формальное определение параметризованных модели Крипке таково. Пусть заданы конечные множеств $A = \{a_1, a_2, ..., a_n\}$ атомарных формул (событий) и $P = \{p_1, p_2, ..., p_m\}$ параметров. Запись For_P будем использовать для обозначения множества всех булевых формул над переменными множества P. Конечной параметризованной моделью Крипке сигнатуры (A,P) называется четверка $\tilde{M} = (S, S_0, R, L)$, где S – непустое конечное множество состояний, S_0 – подмножество начальных состояний, $R : S \times S \to For_P$ – параметризованное множество переходов, $L : S \times A \to For_P$ – параметризованная оценка событий. Для каждого набора σ значений параметров P параметризованная модель $\tilde{M}$ определяет обыкновенную модель Крипке $\tilde{M}[\sigma]$, которая называется примером параметризованной модели $\tilde{M}$. В $\tilde{M}[\sigma]$ из состояния v в состояние u имеется переход в том и только том случае, когда формула $R(v,u)$ на наборе σ принимает значение *true*. Значения формул $L(v,a)$ на наборе σ задают оценку событий в состояниях модели $\tilde{M}[\sigma]$. Для сохранения тотальности отношения переходов в каждом примере $\tilde{M}[\sigma]$ на параметризованное отношение переходов R налагается ограничение: для любого состояния u должно выполняться тождество $\bigvee_{v \in S} R(u,v) \equiv true$. Сформулируем задачу верификации конечных параметризованных моделей. Пусть задана булева формула g из множества For_P, используемая для ограничения значений переменных-параметров. Темпоральная формула φ, зависящая от событий A, считается выполнимой в конечной параметризованной модели $\tilde{M}$ при ограничении параметризации g (этот факт будем обозначать записью $\tilde{M}, g \models \varphi$), если для любого набора σ значений параметров и для любого начального состояния s_0 выполняется соотношение $g(\sigma) = true \Rightarrow \tilde{M}[\sigma], s_0 \models \varphi$. Задача верификации заключается в проверке соотношения $\tilde{M}, g \models \varphi$ для заданной конечной параметризованной модели $\tilde{M}$, формулы CTL φ и булевого ограничения g.

3. **Символьный алгоритм верификации конечных параметризованных систем для CTL.** Символьные алгоритмы верификации работают со сложными структурами данных (формулами, уравнениями, логическим схемами и др.), позволяющими компактно описывать конечные множества. Конечную параметризованную модель можно определить набором булевых формул, зависящих от событий, параметров и вспомогательных переменных, используемых для представления состояний модели. Для заданной конечной параметризованной модели Крипке $\tilde{M} = (S, S_0, R, L)$ сигнатуры (A, P), где $|A| = n, |P| = m, |S| = 2^k$, введем множество вспомогательных булевых переменных $Z = \{z_1, z_2, ..., z_k\}$ и сопоставим взаимно однозначно каждому состоянию s из множества S набор Δ_s значений вспомогательных переменных. Тогда конечная параметризованная модель $\tilde{M}$ полностью определяется набором, состоящим из n+2 булевых формул $\left\langle f_0^{(k)}, f_R^{(m+2k)}, f_1^{(m+k)}, f_2^{(m+k)}, ..., f_n^{(m+k)} \right\rangle$, удовлетворяющих условиям:

1) $s \in S_0 \Leftrightarrow f_0(\Delta_s) = true$;

2) для любой пары состояний u, v верно соотношение $f_R(\Delta_u, \Delta_v, p_1, ..., p_m) \equiv R(u, v)$;

3) для любого состояния u справедливо соотношение $f_i(\Delta_u, p_1, ..., p_m) \equiv L(u, a_i)$;

Верификация параметризованной модели $\tilde{M}$ проводится в два этапа. На первом этапе для заданной CTL формулы φ вычисляется (m+k)-местная булева формула $h_\varphi(\bar{z}, \bar{p})$, которая характеризует выполнимость темпоральной формулы φ в каждом состоянии системы в зависимости от значений параметров. Как известно (см. [1]), каждая CTL-формула равносильна темпоральной формуле, в которой применяются только булевы связки $\wedge, \neg$ и темпоральные операторы $\exists X, \exists U, \exists G$. Поэтому достаточно представить процедуры вычисления булевых формул $h_\varphi^M(\bar{z}, \bar{p})$ для темпоральных формул φ, озаглавленных указанными логическими связками и операторами.

1. Если $\varphi = a_i$, где $a_i \in A$, то $h_\varphi^M(\bar{z}, \bar{p}) \equiv f_i$.
2. Если $\varphi = \neg\psi$, то $h_\varphi^M(\bar{z}, \bar{p}) \equiv \neg h_\psi^M(\bar{z}, \bar{p})$.
3. Если $\varphi = \psi \wedge \chi$, то $h_\varphi^M(\bar{z}, \bar{p}) \equiv h_\psi^M(\bar{z}, \bar{p}) \wedge h_\chi^M(\bar{z}, \bar{p})$.
4. Если $\varphi = \exists X\psi$, то $h_\varphi^M(\bar{z}, \bar{p}) = \exists \bar{z}_0 (f_R(\bar{z}, \bar{z}_0, \bar{p}) \wedge h_\psi^M(\bar{z}_0, \bar{p}))$.
5. Если $\varphi = \exists(\psi U \chi)$, то h_φ^M – это булева функция, которая является наименьшим в решетке булевых функций решением функционального уравнения
$$Y(\bar{z}, \bar{p}) = h_\chi^M(\bar{z}, \bar{p}) \vee (h_\psi^M(\bar{z}, \bar{p}) \wedge \exists \bar{z}_0 (f_R(\bar{z}, \bar{z}_0, \bar{p}) \wedge Y(\bar{z}_0, \bar{p}))).$$
Это решение может быть вычислено следующей процедурой:
Y=0; Y′=h_χ; **while** Y ≠ Y′ **do** Y:= Y′; Y′:=$h_\chi \vee (h_\psi \wedge \exists z_0 (f_R \wedge Y))$ **od.**
6. Если $\varphi = \exists G\psi$, то h_φ^M – это булева функция, которая является наибольшим в решетке булевых функций решением функционального уравнения
$$Y(\bar{z}, \bar{p}) = h_\psi^M(\bar{z}, \bar{p}) \wedge \exists \bar{z}_0 (f_R(\bar{z}, \bar{z}_0, \bar{p}) \wedge Y(\bar{z}_0, \bar{p})).$$
Это решение может быть вычислено следующей процедурой:
Y=1; Y′=h_ψ; **while** Y ≠ Y′ **do** Y:= Y′; Y′:= $h_\psi \wedge \exists z_0 (f_R \wedge Y)$ **od.**

Здесь 1 и 0 обозначают булевы функции, которые тождественно равны *true* и *false* соответственно Применение квантора существования $\exists z f(x, z)$ к булевой формуле представляет собой сокращенную запись формулы $f(x,0) \vee f(x,1)$.

Лемма 1. Для любой конечной параметризованной модели Крипке $\tilde{M}$, CTL-формулы φ и набора значений параметров σ верно соотношение $\tilde{M}_\sigma, s \models \varphi \Leftrightarrow h_\varphi^M(\sigma) = true$.

Эта лемма является обобщением аналогичного утверждения для обычных моделей Крипке (см. [1]).

На втором этапе верификации параметризованной модели $\widetilde{M} = \left\langle f_0^{(k)}, f_R^{(m+2k)}, f_1^{(m+k)}, f_2^{(m+k)}, \ldots, f_n^{(m+k)} \right\rangle$ относительно темпоральной формулы φ и ограничения параметризации g проводится проверка общезначимости следующей формулы

$$f_{M,\varphi,g} = (g \wedge f_0 \to h_\varphi^M).$$

Непосредственно из леммы 1 вытекает

Теорема 1. Для любой конечной параметризованной модели Крипке $\widetilde{M}$, CTL-формулы φ и ограничения параметризации g верно соотношение $\widetilde{M}, g \models \varphi \Leftrightarrow f_{M,\varphi,g} \equiv true.$

Описанный здесь алгоритм верификации конечных параметризованных моделей может быть реализован на практике с привлечением пакетов построения и преобразования упорядоченных двоичных разрешающих диаграмм (OBDD) и процедур проверки выполнимости булевых формул (SAT-solvers).

4. **Сложность верификации конечных параметризованных моделей.** Как известно, проверку выполнимости CTL-формулы длины n в обыкновенной модели Крипке, имеющей m переходов, можно провести за время $O(nm)$. Однако для параметризованных моделей Крипке сложность задачи верификации значительно возрастает. Обнаружено, что это происходит даже в том случае, когда булевы формулы $R(u,v)$ и $L(u,a)$, помечающие переходы между состояниями и оценку событий, не содержат логических связок, т.е. являются либо булевыми параметрами, либо булевыми константами. Конечную параметризованную модель Крипке $\widetilde{M} = (S, S_0, R, L)$ назовем простой моделью, если для любой пары состояний s', s'' и для любого события a верны включения $R(s', s'') \in P \cup \{0,1\}$ и $L(s', a) \in P \cup \{0,1\}$. Далее будет показано, что задача верификации простых конечных параметризованных моделей Крипке является вычислительно трудной. Для этого предложен следующий метод сведения проблемы невыполнимости конъюнктивных нормальных форм (КНФ) к задаче верификации простых параметризованных моделей.

Пусть задана произвольная 3-КНФ C над множеством переменных $X = \{x_1, x_2, \ldots, x_m\}$. Введем вспомогательные булевы переменные $x_1', x_2', \ldots, x_m'$ и заменим в булевой формуле C все отрицательные литеры $\neg x_i, 1 \le i \le m,$ соответствующими штрихованными переменными x_i'. В результате этой замены будет получена 3-КНФ

$$C' = (z_{11} \vee z_{12} \vee z_{13}) \wedge (z_{21} \vee z_{22} \vee z_{23}) \wedge \ldots \wedge (z_{N1} \vee z_{N2} \vee z_{N3}),$$

в которой содержатся только положительные литеры. В качестве множества событий выберем множество $A = \{a, b, c, d\}$ и рассмотрим простую конечную параметризованную модель Крипке $\widetilde{M}^C$, изображенную на рисунке. В этой модели для упрощения записи все непомеченные стрелки соответствуют переходам с пометкой 1, а отсутствие стрелки, соединяющей пару различных состояний, соответствует переходу с пометкой 0. Кроме того, для соблюдения тотальности отношения переходов предполагается, что из каждого состояния модели исходит переход, ведущий в то же самое состояние. В каждом состоянии модели $\widetilde{M}^C$ истинными считаются те и только те события из множества A, которые приписаны состояниям модели (см. рис.).

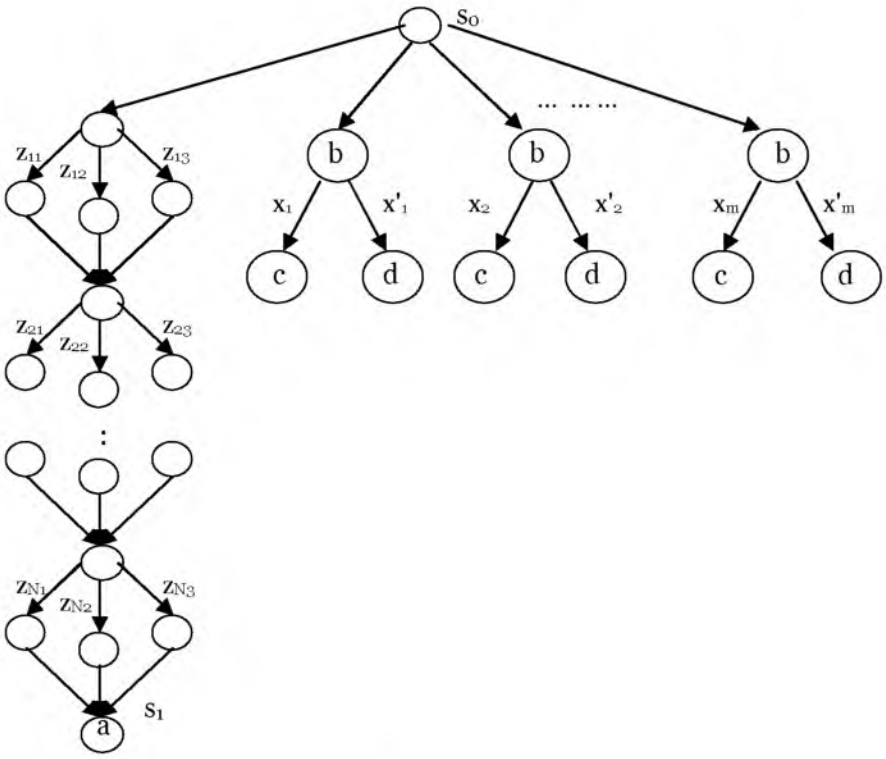

Рис. Модель Крипке $\widehat{M}^C$, соответствующая 3-КНФ C

Лемма 2. Для любой 3-КНФ C справедливо соотношение

$$\widehat{M}^C, true \models \forall G(b \to (\exists Xc \oplus \exists Xd)) \to \forall G \neg a \quad \Leftrightarrow C \text{ – выполнимая 3-КНФ.}$$

Здесь значки $\to$ и $\oplus$ обозначают логические связки импликации и исключающего ИЛИ.

Доказательство. Невыполнимость $\widehat{M}^C, true \models \forall G(b \to (\exists Xc \oplus \exists Xd)) \to \forall G \neg a$ означает, что для некоторого набора δ значений параметров $x_1, \ldots, x_m, x'_1, \ldots, x'_m$ верны соотношения $\widehat{M}^C[\delta], s_0 \models \forall G(b \to (\exists Xc \oplus \exists Xd))$ и $\widehat{M}^C[\delta], s_0 \models \neg \forall G \neg a$. Первое из них равносильно тому, что для каждой пары переменных $x_i, x'_i, 1 \le i \le m$, имеет место равенство $x'_i[\delta] = \neg x_i[\delta]$. Так как в CTL формула $\neg \forall G \neg a$ равносильна $\exists Fa$, второе из указанных соотношений означает, что в модели Крипке $\widehat{M}^C[\delta]$ имеется вычисление, ведущее из состояния s_0 в состояние s_1. Учитывая устройство $\widehat{M}^C[\delta]$ последнее означает, что $C'[\delta] = true$. Таким образом, указанные выше соотношения выполнимости равносильны тому, что $C(x_1[\delta], x_2[\delta], \ldots, x_m[\delta]) = true$, т.е. C – выполнимая 3-КНФ.

Из леммы 2 вытекает следующая теорема, дающая оценку сложности задачи верификации конечных параметризованных моделей в логике CTL.

Теорема 2. Проблема невыполнимости CTL-формул в простых параметризованных моделях Крипке является NP-полной.

Литература

1. Кларк Э.М., Грамберг О., Пелед Д. Верификация моделей программ (model checking). М.: МЦНМО, 2002. – 416 с.
2. Cimatti A., Clarke E., Giunchiglia F., Roveri M. nuSMV: a new symbolic model verifier. In Proceedings of CAV'99 (Computer Aided Verification), 1999, p. 495–499.
3. Holzmann G.J. The SPIN Model Checker: Primer and Reference Manual. Addison-Wesley, 2003, 608 p.

4. Apt K.R., Kozen D. Limits for automatic program verification of finite-state concurrent systems. Information Processing Letters, v. 22, N 6, 1986, p. 307–309.
5. Clarke E.M., Filkorn T., Jha S. Exploiting symmetry in temporal logic model checking. Proceedings of CAV'93, Lecture Notes in Computer Science, v. 697, 1993, p. 450–461.
6. Emerson E.A., Namjoshi K.S. Reasoning about rings. Proceedings of the 22th ACM Conference POPL'95 (Principles of Programming Languages), 1995, p.85–94.
7. Emerson E.A., Sistla A.P. Symmetry and model checking. Formal Methods in System Design, v. 9, N 1/2, 1996, p.105–131.
8. Donaldson A.F. Miller A. Automatic symmetry detection for model checking using computational group theory. Proceedings of the 13th International Symposium on Formal Methods Europe (FME 2005)}, Lecture Notes in Computer Science, v. 3582, 2005, p. 481–496.
9. Коннов И.В., Захаров В.А.. Об одном подходе к верификации симметрических параметризованных распределенных систем // Программирование. – 2005 -№ 5.
10. Clarke E.M., Grumberg, O., and Jha, S. Verifying parameterized networks using abstraction and regular languagesю Proceedings of the 6-th International Conference on Concurrency Theory, 1995.
11. Dams D., Grumberg O., Gerth R. Abstract interpretation of reactive systems: abstractions preserving ACTL*, ECTL* and CTL*. IFIP Working Conference and Programming Concepts, Methods and Calculii, 1994.
12. Kesten Y., Pnueli A. Verification by finitary abstraction. Information and Computation, v. 163, 2000, p.203–243.
13. Kurshan R.P., MacMillan K.L. Structural induction theorem for processes. Proceedings of the 8-th International Symposium on Principles of Distributed Computing, PODC'89, 1989, p. 239–247.
14. Wolper P., Lovinfosse. Properties of large sets of processes with network invariants. Lecture Notes in Computer Science, 407, 1989, p. 68–80.
15. Calder M., Miller A. Five ways to use induction and symmetry in the verification of networks of processes by model-checking. Proceedings of AvoCS 2002 (Automated Verification of Critical Systems), 2002, p. 29–42.
16. Редькин Н.П. Надежность и диагностика схем / М.: Изд-во МГУ, 1992.
17. Godefroid P., Bruns G. Generalized model checking: reasoning about partial state spaces. In Proc. of CONCUR 2000, p. 168–182.

ON THE VERIFICATION OF FINITE STATE PARAMETERIZED MODELS OF DISTRIBUTED PROGRANS

P.E. BULYCHEV
V.A. ZAKHAROV

Lomonosov Moscow State University

e-mail: zakh@ cs.msu.su

In this paper the authors introduced and studied a new class of parameterized models for distributed programs. These models are finite state transition systems that involve Boolean variables as parameters. Boolean parameterization makes it possible to specify succinctly finite families of communicating processes whose behavior is synchronized in some aspects. We introduced also a symbolic model checking algorithm for verification of finite state parameterized models of distributed programs against CTL specifications. The complexity of verification problem is also estimated.

Keywords: program, transition system, specification, temporal logic, verification, Boolean function, OBDD, NP-complete problem.